Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2020-372-RC1, 2021

Geoscientific
Model Development
Discussions

Interactive comment on "LARGE 0.2.0: 2D numerical modelling of geodynamic problems" *by* Nicola Creati and Roberto Vidmar

John Mansour (Referee)

john.mansour@monash.edu

Received and published: 10 February 2021

General comments:

Hello Nicola & Roberto. My name is John Mansour and I'm part of the Underworld development team. I'm more a numerical developer than geophysics modeller, and this review will reflect that.

The authors gives an overview of the LARGE Python geodynamics modelling software. LARGE uses a 2D finite difference method with Lagrangian particles to track material evolution and can operate in serial or MPI parallel. It provides methods for incompressible Stokes solutions, and advection-diffusion solution. LARGE provides a rheological model for visco-elasto-plastic behaviours for which the user can set a range

of parameters as necessary. The authors of LARGE should be commended for being quite usability minded, an often overlooked aspect of software design. They achieve this through features such as easy documentation access, a GUI and visualisation tool, logging tools, and a framework for ensuring consistent units usage.

While LARGE appears to be a quality package and will be a welcome addition to the Python geoscience ecosystem, I believe there are some shortcomings in this paper that need to be overcome before it can be recommended for publication. The biggest issue is that the authors haven't given any evidence to support the fidelity of their software. A number of models are presented, but all I can say about these models is that they look qualitatively reasonable. However, we are almost completely in the dark with respect to quantitative accuracy (they do mention shear band angles being reasonable, but that is all). Things that should be done include: * Resolution tests to show behaviour is convergent. * Comparison with analytic tests, and convergence order analysis. * Comparison with results from the literature.

For the analytic tests, Underworld2 provides a suite of tests to ensure that the Stokes system is generating consistent results. SolCx in particular is a standard test most geodynamics software papers will demonstrate. Here are some examples:

https://github.com/underworldcode/underworld2/blob/master/docs/examples/10_Analytic%2(
https://github.com/underworldcode/underworld2/blob/master/docs/test/Analytic%20Soln%20

Some published results you might consider comparing with: - Blankenbach et al (1989), 2D thermal convection - vanKeken et al (1997), 2D Rayleigh-Taylor and thermochem evolution - Moresi & Solomatov (1998), 2D mantle convection with a brittle lithosphere - Crameri et al (2012), surface topography, sticky air. - Tosi et al (2015), 2D viscoplastic thermal convection - Farrington et al (2014), Viscoelastic shear. - Duretz et al (2011) and Schmid & Podladchikov (2003), circular inclusion - Lemiale et al (2008), shear banding in plastic models - OzBench (2008), slab subduction.

The other main issue with this paper is that the English is too casual. While I understand that English is not the first language of the authors, and I have provided some alternate phrasing suggestions in places, perhaps finding someone to proof read would help expedite this process. There are also simple spelling mistakes that a spell checker should have picked up.

As outlined below, I believe this paper would also be well served to always give the reader enough information to have a basic idea about the different algorithms chosen to be used within LARGE, instead of only providing a reference and assuming that the reader is familiar with that work. The reader shouldn't *need* to access that reference to have basic understanding of what the algorithm entails.

I have outlined a bunch of further smaller issues in the sections below, but one other thing I'll mention is that you should let us know what are the skill requirements necessary to use LARGE. For example, do users need to know Python?

I hope this has been helpful and look forward to seeing the next iteration of this paper. John

Scientific Questions/Issues:

L4-5: "LARGE uses advanced modern numerical libraries and algorithms but unlike common simulation code written in Fortran or C this code is written entirely in Python." I'd be a bit more careful about the wording here. While LARGE is written entirely in Python, most of the libraries LARGE uses will be written in C/Fortran/etc. Also, you should mention some of the other Python based geodynamics codes currently available (UWGeodynamics/Underworld2/PyLith/etc) and how LARGE differs from these.

L18-20: "Geodynamic simulation can be accomplished using commercial software (Pascal and Cloetingh, 2002; Jarosinski et al., 2011), but there are also several other tools available in the literature, both open and closed source, that do the same job and add new numerical methods and algorithms." I think the point to make about open source is that it is good for open science because it provides transparency into the

underlying implementations, and also reproducibility because people can easily re-run the computational experiments (which might not be the case for closed-source and/or commercial software).

L29: Python is perfect suited to large datasets too! Not just small and medium. Check 'dask' for example.

L33-34: "The language is self-sufficient, comes out-of-the-box ready to use, with everything that is needed." What does this mean? I'm not sure it's a useful statement.

L39-50: I'd suggest simplifying this section greatly. I think your target audience should be very well aware of most/all of this. You might simply say "LARGE has been developed on Linux desktop computers, although for larger simulations it may also be deployed to run across multi-node HPC systems."

l50: You make some claims here about Python operation in parallel. Perhaps you can provide some examples to clarify what you are saying here, and compare with other languages for contrast.

l66-67: I think you need to be a bit more specific here. Which finite difference scheme? What order is it? Forwards/backwards/centred/etc? You provide references which is good, but I think your readers shouldn't need to dive into those references to have a clear idea what you're doing. Similarly, what does a staggered mesh mean? What do you mean by regular or irregular? Perhaps an diagram here wouldn't hurt to spell out more succinctly how your discretisation works.

l67-68: Similarly, you'll need to spell out how you are treating the energy equation. Is it a Semi-Largrangian approach, or do your temperature unknowns truly advect freely with the flow?

l73-74: Again, give your reader at least a quick overview of what CorrMinMod is all about, and why it is useful for the class of models you are targeting.

l96-97: "where F is a dimensionless correction term that depends on the type of experiment used to measure the rock rheological parameters and lets to transform the equation in tensorial format". Can you clarify what you are saying here?

l103: Again, specify which iterative scheme you are using to handle the non-linearity. Picard? Newton?

l108: It doesn't follow from unconditional stability that iteration isn't required, although unconditional stability allows for arbitrary time-step sizes.

l110: Again, you need to provide enough detail for your readers to understand what 4-cell or 1-cell is. This isn't standard nomenclature, so if you include it you'll need to explicitly explain what it is. Further, this isn't a bilinear-interpolation scheme. It is a weighted averaging scheme which relies on bilinear-shape functions to determine the weighting.

l132: I'd probably leave details about decorators out, as it's a bit Python language specific, and non-Python readers won't have any idea what it's about.

l135: ‘colorAlphabet‘ seems a bit out of place here as it's not really a scientific concern, *unless* you go on to talk about usability aspects of LARGE and how this is helps to make model development easier.

l154: I'm not sure I'm convinced that this should be listed as distinct from h5py. If it's simply h5py bundled with the necessary libhdf5 libraries, then as far as the paper is concerned it's just h5py.

l189: Give a quick overview of the Halton distribution and why it's used. Diagrams are always nice.

l206: You should give a reference for PETSc. I'd also be interested in the performance differences between the SciPy solver and the PETSc solver (in serial), but perhaps you'll touch on this later.

l219-221: Can you clarify this sentence? Does it really become ill-posed (as in, no

solution exists), or does convergence simply become difficult? What do you mean by "iterating on the domain grid nodes"? l234: I'd probably suggest avoiding speaking about specific parameters (such as "tracersReseedGlobal") as these are more appropriate for your user documentation, and are also quite likely to change as LARGE evolves.

l239: Similarly, perhaps better to omit Figure 3, as these details are almost certain to change in newer versions of LARGE. It's enough to say what you've said l238-240

l244: I'm a bit unsure what you're saying here. Perhaps something like this is sufficient? "LARGEP uses an MPI friendly logging system that minimised user difficulty during the model development stage."

l246-268: Command line utilities: I'd be inclined to think a lot of the details in the section are best left for a LARGE user manual or other similar documentation. I think it's fine to speak about the GUI, and other tools that you have generated to assist the user experience, but discussion should be centred on why it is important for the user experience, rather than the specific tool itself. For example, "To facilitate rapid model development, user documentation for LARGE is made readily available browseable HTML pages proving reference for configuration file specification, Python API, examples, etc etc". Some of the nitty gritty (such as ‘large2largep‘, ‘largep‘) isn't really appropriate here I think, so think to simplify as much as possible.

l270+: Examples. Again, I think you should omit details about configuration files and parameter specifications. This paper should be more focused on the science, not the API. So for example you say: "This simulation does not take in account the temperature therefore the heat calculation is switched off in the configuration file by setting the equation solver to none." where you probably should simply say "This simulation does not take into account temperature effects".

l273: Shear bands. How does your 26 degree shear band angle compare to the results of Kaus (2010)? I believe Moresi et al (2007) also performs similar numerical experiments. Are these results convergent with respect to resolution? What resolution were the simulations performed at? How many particles?

l295: Plates converge: What resolution was used? Particle counts? How does this compare to the results of Gorczyk et al 2007? It looks qualitatively reasonable, but why should we believe this?

l314: Slab retreat: Same questions.. resolution/particles? Comparison with other results? There's no shortage of similar models run using Underworld1/Underword2/other. We want to know whether your results are in line with other published results, and if not then why.

l330: Continental Rifting. Again, it all looks reasonable, but you'll need to convince us that it should be believed.

l364: Which parallel packages? Petsc4py? Anything else? Perhaps specify where most of the parallel time is going.

l370: Fit to this curve will only be true if we assume perfect parallel operation (for the non-serial parts of operation). Of course, this is never truly going to occur due to communicational overheads etc. It might be useful to provide a table or graph showing how execution time is spread across the different simulation phases, and show which phases are serial and which are parallel. Also, i'd be interesting to know how LARGE compares to LARGEP when LARGEP is running using a single processor.. ie PETSc solves vs SciPy solves. Also, even laptops these days have 4+ cores.. how does running on a laptop/desktop utilising multiple cores compare to serial operation?

Technical Corrections:

L5-6: Suggestion: "Simulations are driven by configuration files that define the lithologies and parameters for a given model."

L6-8: Suggestion: "The package can be used to reproduce results of published studies and models or to experiment with new simulations. LARGE can run in serial mode on

desktop computers but can also take advantage of MPI to run in parallel on multi node HPC systems."

L13: Full stop (not comma) required after "measured".

L22-23: Suggestion: "All these tools are written either in Fortran, C or C++ and integrate with well established libraries that provide numerical methods for the solution of differential equations"

L29-31: Suggestion: "The main reasons for Python's success are that it is simple, free, interoperable with other languages, object oriented, and has a large scientific community of users and developers continually extending its potential capabilities."

L31-32: Suggestion: "Since its introduction, several packages have been developed to improve its numerical computation capabilities, in particular Numpy and SciPy (Walt et al., 2011; Virtanen et al., 2020).

L34-35: Suggestion: "Being an interpreted language, it is inherently slow relative to compiled languages, but this limitation can be avoided with careful implementation of the required algorithms."

l100-101: Suggestion: "... the enucleation of shear deformation zones."

l101-102: Suggestion: "The Stokes equations is non-linear and the solution process is not guaranteed to converge (Spiegelman et al., 2016). The non-linearity is owing to the deviatoric stress (eqn 2) which is dependent on a viscosity coefficient which is itself dependent on the velocity and/or pressure solutions."

l108: Cranck -> Crank

l120: Suggestion: "It is the standard implementation for numerical array structures in Python, and its universal adoption greatly enhances interoperability within the scientific Python ecosystem. Leveraging Numpy's vectorized methods in place of standard Python for-loop constructs allows for run-time execution which largely avoids the

Python interpreter. This provides dramatic improvements in numerical throughput, although code readability is at times impacted.

l169-170: Suggestion: "LARGE supports the most commonly used boundary conditions within the geodynamics literature. The Stokes system may be configured for free-slip and no-slip conditions."

l180: "nodes density" -> "node density"

l190-191: Suggestion: "Utilising Numpy and MPI (mpi4py?), this object takes handles all communication between the processors for tracer distribution, advection, deletion, repopulation and ghosting."

l270-272: Suggestion: "The ability of LARGE to solve geodynamics problems can be tested by running standard community models and comparing with published results from within the literature. These models are also provided with LARGE for users to familiarise themselves with the LARGE modelling paradigms."

l303: "Stikcy" -> "Sticky"

l342: "Stikcy" -> "Sticky"

l364: "... but parallel scalability of LARGE is largely dependent on the scalability of the third party dependencies utilised for parallel operation."

l368: Suggestion: ".. promising" -> ".. encouraging".

l386: "weel" -> "well"

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2020-372, 2020.